

# **Rapport final**

**SEG2505 - Introduction au génie logiciel**

**2024**

**Université d'Ottawa**

Professeur : Laurent FREREBEAU

Groupe 33

Nicola Baker, 300360908

01/12/2024

## Table des matières

1. Introduction .....	3
2. Description de mes Choix de Conception .....	3
3. Description des exigences supplémentaires proposées, Traitées ou Écartées .....	4
4. Retour d'expérience sur la gestion du projet.....	6
5. Tableau de synthèse des contributions.....	7
6. Estimation du temps global .....	8
7. Conclusion.....	8

Le lien : <https://github.com/uOttawa-2024-2025-seg2505-projet/groupe-33>

## **1. Introduction**

Dans le cours SEG2505, j'ai travaillé sur un projet visant à concevoir et développer une application Android qui permet de commander des PC sur mesure. L'objectif était de créer un système simple, où différents utilisateurs tels que des Requesters, un StoreKeeper, un Assembler et un Administrateur, pourraient interagir avec l'application selon leur rôle. L'application devait intégrer une base de données en utilisant Firebase Firestore afin de gérer les utilisateurs, les commandes et le stock de composants en temps réel. En plus de la partie développement le projet impliquait de modéliser l'architecture à l'aide de diagrammes UML et de livrer l'application dans des délais bien définis. Ce projet a été une excellente occasion de mettre en pratique mes compétences en développement mobile en gestion de données, et en conception logicielle.

## **2. Description de mes Choix de Conception**

Mon approche de conception a été de rendre l'expérience aussi fluide que possible, en prenant en compte l'accessibilité et la facilité de navigation. Voici un aperçu de mes choix de conception.

### **Structure de l'interface**

J'ai divisé l'interface en deux grandes sections de Composants:

- Composants matériels : cela inclut le boîtier, la carte mère, les barrettes mémoire, les disques durs, les écrans et le clavier-souris.
- Composants logiciels : incluant le navigateur web, la suite bureautique, et les outils de développement.

Chaque section est bien délimitée et clairement étiquetée avec des titres afin que l'utilisateur sache exactement ce qu'il est en train de choisir.

Pour chaque composant matériel, j'ai ajouté un champ de saisie EditText pour que l'utilisateur puisse spécifier la quantité. Par exemple, il peut choisir combien de barrettes mémoires ou d'écrans il veut. J'ai utilisé un champ de texte numérique pour s'assurer que seules des valeurs numériques soient entrées. Ce choix permet à l'utilisateur de définir exactement ses besoins et évitant des erreurs de saisie.

### **Interactions Utilisateur**

- Spinners : J'ai utilisé des Spinners pour permettre à l'utilisateur de choisir parmi les différentes options pour chaque composant (comme le type de boîtier ou la carte

- mère, etc..). Les utilisateurs peuvent faire un choix parmi les options disponibles grâce à cette interface simple et déroulante.
- RecyclerView pour l’affichage des commandes : Pour afficher les commandes, j’ai utilisé un RecyclerView, car il est performant et personnalisable. Il permet de gérer efficacement une liste de commandes même si elle devient grande. Cela rend l’interface fluide et facile à naviguer.

### **Choix de Conception accessibilité**

J’ai choisi Firebase Firestore comme base de données, car elle est facile à utiliser et gère les données en temps réel ce qui me permet de stocker chaque commande sous forme de document avec des champs. Cette structure flexible m’a aidé à éviter les relations compliquées entre les données.

J’ai intégré des messages d’erreur pour informer l’utilisateur si quelque chose ne va pas, par exemple, si la mise à jour de l’état échoue. Cela aide à éviter la confusion et rend l’application plus fiable.

L’application est construite de manière modulaire ce qui rend les futures modifications plus faciles. Par exemple, je peux ajouter des fonctionnalités comme des filtres ou une recherche de commandes sans devoir refaire tout le code.

Ces choix de conception m’ont permis de créer une application pratique, en assurant une bonne expérience utilisateur et une gestion efficace des données.

## **3. Description des exigences supplémentaires proposées, Traitées ou Écartées**

Dans le cadre du développement de mon application qui permet de commander des PC sur mesure, plusieurs exigences supplémentaires ont été mises en place pour améliorer l’interface utilisateur, la gestion des commandes. L’objectif était de rendre l’application plus fonctionnelle. Certaines de ces exigences ont été mises en œuvre tandis que d’autres ont été écartées ou simplifiées pour assurer une meilleure expérience utilisateur.

L’une des premières exigences était de permettre à l’utilisateur de voir facilement les composants et les commandes sur une seule page. Pour cela, j’ai utilisé un RecyclerView. Cette approche permet de lister tous les composants (boîtier, carte mère, barrettes mémoire,

disques durs, écrans, etc.) et commandes de manière efficace. Le RecyclerView permet à l'utilisateur de voir les produits choisis. Cette fonctionnalité améliore l'expérience utilisateur en rendant l'interface plus simple à utiliser, même avec une grande quantité de données.

J'ai ajouté une fonctionnalité d'historique des commandes rejetées et livrées permettant à l'Assembleur de consulter le statut de ses commandes passées. Les commandes rejetées indiquent la raison du rejet tandis que les commandes livrées sont confirmées avec un statut de livraison. De plus, il est possible de restaurer une commande rejetée en reprenant la commande offrant ainsi plus de flexibilité à l'Assembleur.

Afin de faciliter le choix des composants j'ai utilisé des Spinners pour chaque type de composant. Ces Spinners permettent à l'utilisateur de sélectionner les composants (boîtier, carte mère, disque dur, etc.) parmi une liste d'options prédéfinies. L'avantage des Spinners est qu'ils permettent de gagner de l'espace dans l'interface en offrant une grande variété d'options à l'utilisateur. Cela rend l'application plus claire et réduisant la possibilité d'erreurs de saisie. L'utilisateur peut facilement sélectionner les composants nécessaires sans que l'interface ne devienne trop encombrée.

Un autre aspect a été la gestion des rejets lors du traitement des commandes. Pour rendre l'expérience utilisateur plus fluide, j'ai ajouté un système de messages personnalisés. Par exemple, si une commande est rejetée, l'utilisateur reçoit un message qui lui explique pourquoi. Ce message peut être de deux types :

- Option 1 : Un message général qui explique qu'il y a un problème avec la commande.
- Option 2 : Un message plus détaillé qui précise ce qui manque dans le stock pour compléter la commande.

Cela aide l'utilisateur à comprendre la situation et à ajuster sa commande en fonction des informations données.

Certaines fonctionnalités qui étaient envisagées au départ ont été écartées pour simplifier l'application et éviter une surcharge de fonctionnalités qui risquait de rendre l'interface trop compliquée. Par exemple, l'idée d'ajouter une fonctionnalité pour personnaliser chaque composant avec des informations détaillées, des images a été mise de côté. Bien que cette fonctionnalité puisse enrichir l'expérience elle aurait rendu l'interface trop lourde et moins pratique à utiliser.

## **4. Retour d'expérience sur la gestion du projet**

### **Organisation de l'équipe**

Bien que j'aie réalisé le projet seul, j'ai cherché à appliquer une organisation similaire à celle d'une équipe. Pour cela, j'ai utilisé des outils comme To Do pour planifier et suivre les tâches à réaliser. Chaque semaine, je prévoyais mon travail à l'avance pour m'assurer de respecter les délais. J'ai adopté une approche proactive en accomplissant mes tâches une semaine à l'avance. Cela m'a permis de vérifier et ajuster mon travail dans la dernière semaine avant le livrable, et ainsi éviter toute précipitation de dernière minute.

Durant ma semaine de relâche, j'ai concentré mes efforts sur les fonctionnalités utilisateur afin de me libérer du travail pendant les semaines les plus intensives. Cette approche m'a permis de gérer efficacement mon temps même en l'absence d'un partenaire de projet et d'assurer une progression continue vers le livrable final.

### **Difficultés rencontrées et solutions apportées**

Le projet n'a pas été sans défis. L'un des premiers obstacles a été la réalisation des diagrammes UML. J'ai essayé de les créer dès la première semaine, mais n'ayant pas encore vu les notions en classe j'ai rencontré de grandes difficultés. J'ai dû me fier à des recherches personnelles pour comprendre comment représenter correctement les relations entre les classes, les états et les séquences. Malgré mes efforts, je n'étais pas entièrement sûr. Ce n'est qu'après les cours portant sur les diagrammes UML quelques semaines plus tard que tout s'est éclairci. J'ai pu revoir mes diagrammes avec une meilleure compréhension et les ajuster selon les choses apprises en classe.

L'une des difficultés rencontrées a été la prise en main de Firebase au début du projet. Bien que j'aie déjà eu quelques expériences avec Firebase avant de suivre des tutoriels, il m'a fallu un certain temps pour comprendre son fonctionnement et déterminer quelles fonctionnalités pouvaient répondre à mes besoins. Pour résoudre ce problème, j'ai effectué plusieurs recherches pour mieux comprendre la structure de Firebase ses services et comment les intégrer dans mon application. Cela m'a permis de réussir à implémenter Firebase correctement dans le projet.

Enfin, la gestion du temps a été un défi. Travailler seul signifiait que je devais équilibrer efficacement toutes les tâches, de la conception à la mise en œuvre en respectant les délais. Pour m'en sortir, j'ai adopté une planification stricte, en travaillant une semaine à l'avance.

Cela m'a permis de disposer de temps supplémentaire pour les révisions et les ajustements. Malgré cela certaines étapes comme la rédaction de la documentation ou l'amélioration de l'interface auraient pu avoir de plus de temps si j'avais eu une personne supplémentaire pour partager la charge de travail.

### **Leçons pour un futur projet**

Si ce projet avait été réalisé en équipe cela aurait sans doute facilité certaines étapes, en ce qui concerne la rédaction et la réalisation des diagrammes UML. Bien que j'ai pu répondre à toutes les exigences et livrer mon travail à temps avoir un partenaire m'aurait permis de mieux répartir les tâches. En particulier, cela aurait allégé le travail sur l'interface et les diagrammes et en offrant un soutien pour les phases rédactionnelles.

Une équipe aurait également permis de retours, ce qui aurait peut-être amélioré la qualité du travail dans certains domaines, pour les parties où des choix d'interface ont été faits sans validation externe. Cela m'aurait aussi donné plus de temps pour me concentrer sur des détails spécifiques.

Pour un futur projet, il serait essentiel de bien définir les rôles et responsabilités dès le début, même si l'on travaille seul, afin de s'assurer que toutes les tâches sont couvertes. Il est important de rester flexible et l'aide extérieure pourrait être utile.

De plus, pour de futurs projets, je m'assurerais de mieux équilibrer le temps alloué à chaque tâche. Par exemple, il serait mieux de consacrer plus de temps à l'optimisation de l'interface utilisateur et à l'ajout de fonctionnalités supplémentaires une fois les exigences de base respectées. Enfin, je me concentrerais davantage sur l'automatisation des tests pour une application robuste en réduisant le temps de débogage. Avoir des tests automatisés en place dès le début pourrait améliorer le processus de développement.

## **5. Tableau de synthèse des contributions**

Nom	Numéro d'étudiant	Livrable 1	Livrable 2	Livrable 3	Livrable 4	Livrable 5
Nicola Baker	300360908	100%	100%	100%	100%	100%

Nicola - Dans le cadre du projet, j'ai pris en charge l'ensemble des tâches ainsi une gestion complète et autonome. J'ai conçu et développé l'application Android en Java, la création

des interfaces utilisateur pour chaque rôle (Administrateur, StoreKeeper, Assembler et Requester). J'ai intégré Firebase Firestore pour gérer efficacement les données en temps réel. J'ai aussi faite les diagrammes UML (classes, activité, séquence, état) pour modéliser l'architecture du système. En plus du développement, j'ai rédigé un README détaillé qui explique le fonctionnement de l'application, les étapes d'installation et les valeurs par défaut à utiliser. J'ai également pris en main la gestion du dépôt GitHub, la création de tag pour chaque livrable. Enfin, j'ai réalisé des tests approfondis pour m'assurer du bon fonctionnement de toutes les fonctionnalités et j'ai préparé une démonstration finale. Ce projet m'a permis de démontrer mes compétences dans tous les aspects de la conception, du développement et de la gestion, et en donnant une livraison de qualité.

## **6. Estimation du temps global**

Pour ce projet, j'ai consacré 13 heures au Livrable 1, ce qui comprenait la configuration du dépôt GitHub, la mise en place des utilisateurs et la création de la structure de l'application et j'ai commencé à travailler sur le diagramme de classes. J'ai ensuite passé 18 heures sur le Livrable 2, où j'ai développé les fonctionnalités pour le rôle de StoreKeeper et mis une base de données, comme la gestion du stock de composants. Pour le Livrable 3, j'ai travaillé 23 heures, car il s'agissait de créer et tester le système de commandes pour le rôle de Requester et en complétant les diagrammes de séquence et d'état. Le Livrable 4 m'a pris 25 heures, vu la complexité des fonctionnalités de l'Assembler et les ajustements finaux de l'application et finir les diagrammes UML. Enfin, j'ai passé 3 heures à préparer et réaliser la démonstration finale pour présenter le projet.

## **7. Conclusion**

Ce projet a été une expérience qui m'a permis de mettre en pratique mes compétences en développement logiciel. Travailler seul m'a appris à être autonome et à bien gérer mon temps, mais m'a aussi fait réaliser l'importance du travail d'équipe surtout pour des tâches comme la rédaction ou la modélisation UML, qui demandent du temps et de la réflexion. Malgré les difficultés rencontrées comme la prise en main de Firebase ou la création des diagrammes UML, j'ai su m'adapter et trouver des solutions pour livrer un produit complet. Ce projet m'a non seulement permis de renforcer mes compétences, mais aussi de mieux comprendre l'importance d'une bonne planification, des leçons qui me seront très utiles pour mes futurs projets.